Rapid Model Parameterization from Traffic Measurements

KUN-CHAN LAN
and
JOHN HEIDEMANN
USC Information Sciences Institute

The utility of simulations and analysis heavily relies on good models of network traffic. While network traffic constantly is changing over time, existing approaches typically take years from collecting trace, analyzing the data to finally generating and implementing models. In this paper, we describe approaches and tools that support rapid parameterization of traffic models from live network measurements. Rather than treating measured traffic as a time-series of statistics, we utilize the traces to estimate end-user behavior and network conditions to generate application-level simulation models. We also show multi-scaling analytic techniques are helpful for debugging and validating the model. To demonstrate our approaches, we develop structural source-level models for web and FTP traffic and evaluate their accuracy by comparing the outputs of simulation against the original trace. We also compare our work with existing traffic generation tools and show our approach is more flexible in capturing the heterogeneity of traffic. Finally, we automate and integrate the process from trace analysis to model validation for easy model parameterization from new data.

Categories and Subject Descriptors: H.4.0 [**Information Systems Applications**]: General; I.6.0 [**Simulation and Modeling**]: General

General Terms: Measurement

Additional Key Words and Phrases: Network, Model parameterization, Traffic model

1. INTRODUCTION

Simulations are important for exploring and understanding the complexity of networks. However, it is difficult to simulate and model the Internet due to its

Kun-Chan Lan and John Heidemann are partially supported in this work as part of the SAMAN project, funded by DARPA and the Space and Naval Warfare Systems Center San Diego (SPAWAR) under Contract No. N66001-00-C-8066. Any opinions, findings, and conclusions, or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of DARPA or SPAWAR. John Heidemann is also partially supported as part of the CONSER project, funded by NSF as grant number ANI-9986208.
Authors' address: USC Information Sciences Institute, 4676 Admiralty Way, Suite 1001, Marina Del Rey, CA 90292; email: {kclan; johnh}@isi.edu.

scale, heterogeneity and dynamics [Floyd and Paxson 2001]. Internet traffic is constantly changing over time both in volume and statistical properties, even observed at the same location. It is well-known that network traffic follows daily patterns while traffic changes during the day. There are also larger-scale trends in traffic growth. Kim Claffy et al. [McCreary and Claffy 2000] showed the volume of online game traffic is increasing over the years in the backbone traffic. Recently Zhang et al. [2001] showed that, depending on the particular aspect of constancy (the degree to which the relevant Internet properties hold steady) and the dataset under consideration, the constancy of Internet path properties will start to break at the time scale of hours.

If we fix our interest to a single point of time, the traffic still varies at different places due to the immense heterogeneity of the Internet: the diversity of topology and link properties, different protocol usage and user populations in different networks. For example, the traffic at different websites might be different due to their content differences. The distribution of file size in a trace distribution site like Internet Traffic Archive [Paxson 2000] is not heavy-tailed but instead is bimodal, where small files account for web pages that describe traces and large files for traces themselves. Recently Cao et al. [2001] showed that, due to the lower link utilization and higher degree of multiplexing, the traffic in backbone-style links tends to have higher non-stationarity than traffic in the access links.

Even when we only look at one particular part of the network at a single point of time, network traffic can still show great variations just in terms of direction of flows. For example, inbound traffic and outbound traffic seen at the ingress or egress points of the network typically differs for the same reasons as traffic differs by places; bandwidth asymmetries of up to 10:1 are not uncommon [Asaba et al. 1992].

Rapid and unpredictable change of traffic will threaten to make some research obsolete before it is finished. Some assumptions about traffic mix, topology or protocols might only be valid for less than a few years. However, take today's most widely-used web models as an example, it still takes years from collecting traces and analyzing the data to finally generating and implementing models [Barford and Crovella 1998]. Three stages are involved in this time-consuming process: trace collection, design of traffic model, and model parameterization from measurement. In prior work, these stages have typically been combined, with each new experiment requiring development and parameterization of new models. We instead suggest that a sufficiently powerful model can accurately simulate a wide range of web traffic, and then show how that model can be automatically parameterized.

Furthermore, the existing models are all based on a small set of traces collected from one particular part of the network within some particular time period. Considering the Internet's great technical and administrative diversity and immense variations over time regarding how applications are used, it is not obvious that one can model *his* traffic accurately based on the models derived from measurements taken previously from other parts of the network.

Motivated by the challenge and difficulty of modeling constantly-changing Internet traffic, we have developed methodologies and tools that allow users to quickly parameterize traffic models based on the measurements and generate realistic *contemporary* traffic in their simulations. Our approach does not make any underlying assumption of traffic properties (for example, heavy-tailed distribution for file size/transmission time) and hence is more applicable than existing approaches in coping with the heterogeneous nature of the Internet traffic.

Opposed to traditional trace-replay techniques which typically ignore the fact that traffic frequently reacts to the network's current properties, our approaches focus on characterizing source-level patterns in which the data is sent. We have developed tools and methodologies to support this trace-driven application-level modeling approach for generating synthetic traffic. Our initial studies emphasize two types of traffic, web and FTP traffic, and show that we can accurately generate the simulation model from live data in a *timely* fashion, that allows users to simulate their *current* traffic several times per day. Potential applications of a rapid model parameterization tool will include traffic planning and provisioning, on-line simulation for network control, input to network prediction algorithm and generation of high-speed synthetic traffic [Kamath et al. 2002].

Our work has three primary results. First, we strengthen Floyd and Paxson's arguments by showing that network characteristics not only change over time but also show great variations in other dimensions such as locations and flow directions (Section 4). Second, we propose a methodology for rapidly parameterizing traffic models. This approach employs a trace-analysis tool that infers traffic and topology characteristics, and a CDF-based traffic model that can capture widely varying web traffic (Section 6). Finally, we show how our models can be automatically and rapidly parameterized from traces, allowing a user to quickly instantiate models that represent current, local traffic (Section 5).

2. RELATED WORK

Our work builds on prior work in traffic modeling, trace compaction, workload generation and bandwidth estimation.

2.1 Traffic Modeling

Floyd and Paxson [2001] pointed out, to cope with the constantly changing nature of Internet traffic, it is important to capture the *invariants* of the traffic in modeling the Internet. Our methodology is based on a structural modeling approach which emphasize characterizing the source-level pattern in which data is sent. For most applications, the application-level pattern (such as request/reply patterns in web traffic) in which data is sent, does not react to the network dynamics. In this aspect, we consider our models have captured the application structure invariant in the traffic.

The structure we choose to model user behaviors of web traffic is similar to previous work of Mah [1997] and Crovella et al. [Barford and Crovella 1998; Crovella and Bestavros 1996]. We also adopt Mah's approach in terms of describing traffic based on CDF of real data, which has the advantage of being able to represent arbitrary distribution.

2.2 Trace Compaction

Trace compaction generally refers to the techniques used to retrieve "relevant characteristics" from the trace. In this aspect, we have taken similar approaches as the previous studies of Feldmann et al. [Feldmann 2000] and Smith et al. [2001] in the sense that we also manage to reconstruct application-level statistics (eg. request/response) of web traffic on-the-fly from individual packets captured by the sniffer. However, in Feldmann's work, it requires special hardware and software to be able to extract full HTTP level information. The methodology we adopt to construct a web model is closer to Smith's work where they reconstruct the data exchanges in the HTTP connections based on only the TCP/IP header information. (In fact, we have incorporated part of their codes into our tool for parsing TCP/IP header information.) Additionally, we model path characteristics (hence the resulting models can be directly built into the widely used NS network simulator [Breslau et al. 2000]) and provide more comprehensive validation mechanisms including a wavelet-based analysis. Furthermore, in addition to web traffic that previous work has focused on, we include another dominant traffic in our study namely FTP traffic, and automate the whole process from trace analysis to finally implement and validate the models.

2.3 Workload Generation

Research on Internet workload generation has typically focused on creating generative models based on packet traces of various applications. Several studies has adopted this approach to develop workload generators for web traffic including SURGE [Barford and Crovella 1998], IPB [Mah et al. 1998] and work at RPI [Yuksel et al. 2000]. Their work focused on fitting statistics derived from a set of traces to some widely-used distributions which are then used to generate synthetic traffic workload.

However, first, their approaches from collecting traces, analyzing the data, to finally generating and implementing models take too long, (eg. in Crovella's study [Barford and Crovella 1998], it requires modification of browser codes in order to capture the web-user's browsing behavior) considering the network conditions are constantly changing. Given that Internet traffic is changing constantly, it is generally not applicable to characterize the current traffic simply based on statistics collected years ago from different parts of the network. Second, even if we assume the existence of some universal statistical property (eg. heavy-tail distribution of file size), parameterization is still a non-trivial job for the previous models, which are fairly static.

On the contrary, our approach is capable of parameterizing the traces and generating simulation models in a timely fashion that allows the users to study their *current* traffic. In addition to modeling user/application behavior, our work also manages to estimate path characteristics (namely, delay and bottleneck bandwidth) which are important parameters to drive simulation.

2.4 Bandwidth Estimation

There have been a number of techniques proposed in the area of bandwidth estimation. In general, these techniques can be classified into two groups: single packet and packet-pair techniques. The name refers to the number of packets that are used in a single probe.

Single packet techniques are based on the observation that slower links will take longer to transmit a packet than faster links. If we know how long a packet takes to cross each link, the bandwidth of that link can be calculated. There have been a number of implementations of single packet techniques including Jacobson's [1997] pathchar, clink [Downey 1999], utimer [Cheshire and Baker 1995] and pchar [Mah 1999]. Packet pair techniques are often referred to as packet dispersion techniques. A packet experiences a serialization delay across each link due to the bandwidth of the link. Packet pair techniques send two identically sized packets back-to-back, and measure the difference in time between the packets when they arrive at the destination. All recent research into packet pair techniques include bprobe, cprobe [Carter and Crovella 1996], tcpnanly [Paxson 1999] and the work of Lai et al. [Lai and Baker 1999, 2000, 2001]. The recent packet tailgating technique proposed by Lai and Baker [2000] can be considered a hybrid of both single and packet pair techniques.

The approach we adopt to estimate bottleneck bandwidth is in spirit a combination of Sender Based Packet Pair (SBPP) and Receiver Only Packet Pair (ROPP), as described in Lai and Baker [2001], due to the fact we only take passive measurements at one single point of the network.

3. BACKGROUND

In this section we will describe the dataset we use in this study and two statistical techniques, including wavelet scaling plot and Kolmogorov-Smirnov goodness-of-fit test, that help us validate the models.

3.1 Traces

The data used in our study are from two sources. One was collected on the web server of Internet Traffic Archive (this set of traces will be referred to as "ITA" in this paper). The other was recorded at a 100Mbps Ethernet link connecting the Information Science Institute to the rest of the Internet (referred to as "ISI").

ITA trace was collected using publicly available software *tcpdump*. ISI trace was captured via tcpdpriv [Minshall 1997] utility which anonymizes *libpcap*-format (same format used in *tcpdump*) traces. tcpdpriv can collect traces directly or post-process them after collection using a tool like *tcpdump*. Both traces captured all inbound and outbound traffic but only TCP/IP header information was recorded for reasons like privacy and storage overhead. Note that the traffic volume of ITA trace is significantly lower than that of ISI trace and mainly consists of outbound traffic.

The ITA trace was collected during a 24-hour period starting from 15:20 Nov 6, 2001, and shows obviously bimodal distribution of traffic mix consisting primarily of HTTP and FTP traffic. The ISI traces were collected during six

Table I. Summary of ITA and ISI Traces

Trace	ITA	ISI
Date	Nov 2001	Nov 2001
Duration (hr)	24	42
Total Packets	2.5M	218M
Bytes	2.4G	187G
TCP Packets	2.5M (100%)	143.9M (66%)
Bytes	2.4G (100%)	122G (65%)
UDP Packets	3 (0%)	69.8M (32%)
Bytes	150 (0%)	65G (35%)
HTTP Packets	0.1M (4%)	50M (23%)
Bytes	50M (2%)	71G (38%)
FTP Packets	2.4M (96%)	39M (18%)
Bytes	2.35G (98%)	64G (34%)

one-hour sampling periods each day over a seven-day period starting from Nov 9, 2001. The one-hour sampling periods were chosen somewhat arbitrarily with the intention to capture the variation of traffic between different times of the day.

The typical link utilization during collection period is around 16% to 23% and there is no packet drop in our measurement. For simplicity, in this paper we only show the analysis of two sets of one-hour long ISI data which were collected at different times of the same day. One was recorded starting at 2:00 pm Nov 13 2001 (referred to as ISI-1) and the other was recorded starting at 7:00 pm Nov 13 2001 (referred to as ISI-2). Intuitively, one captures the traffic in a normal business hour and the other shows traffic in after-hours. The details of traces are given in Table I.

3.2 Wavelet Scaling Plot

One of the tools we use for validation, the scaling plot, is a wavelet-based analysis [Abry and Veitch 1998] that utilizes wavelet transform of a time series to study its global scaling property, by which we mean the statistics of the time series viewed at each resolution level or scale, taken as a function of scale. More details of this technique were described in Feldmann et al. [1999] and Huang et al. [2001].

To determine the global scaling property of data, we plot $\log(E_j)$, where E_j is the average energy at scale j, as a function of scale j. The energy level E_j is corresponding to the level of irregularity or burstiness of sampled data. The higher E_j is, the more bursty the traffic is at time scale j. The resulting scaling plot can be used to determine qualitative aspects of the scaling behavior of the underlying time series, and identify highly regular traffic patterns that are well-localized in scale. For example, this wavelet-based analysis can uncover the dominant RTT behavior associated with the packets that make up the measured traffic. For our purpose in this study, we validate our model by comparing its scaling plot against the trace's and see if they qualitatively match closely.

3.3 Kolmogorov-Smirnov Goodness of Fit Test

We use the Kolmogorov-Smirnov goodness of fit test [Massey 1951] to formally determine if two sets of traffic data are significantly different from each other, in addition to visually examining their CDF plots. The Kolmogorov-Smirnov D value is the largest absolute difference between the cumulative distributions of two sets of data. We first compute D value of two data sets and then compare the result to the *critical value* of D. For large number of samples, the critical value at the .05 level significance is approximately $\frac{c}{\sqrt{n}}$, where n is the sample size and c is a constant that is distribution-dependent. For example, if the tested data comes from a normal distribution then $c = 1.36$ [Smirnov 1948] ($c = 1.08$ for exponential distribution [Lilliefors 1969] and $c = 0.874$ for Weibull distribution [Chandra et al. 1981]) If the computed D is less than the critical value then we accept the null hypothesis that the distributions of two data sets are not statistically different from each other. There are two limitations to applying the K-S test to our data. First, we do not make assumptions about the data's distribution, and so we can not directly apply the K-S test since we can not determine c. However, comparison of the absolute value of D is appropriate, and we quantitatively use the most restrictive c (= 0.874) as an approximation to perform the test. In other words, at a 0.05 level significance and for 10000 samples, we will claim two data sets are statistically *different* if the maximum absolute deviation between their cumulative distributions is greater than 0.00874. Second, as reported by previous studies [Paxson 1994; Barford and Crovella 1998], it is difficult to apply a goodness-of-fit test for a large empirical data set (it is well known in the statistics community that large datasets almost never have statistically exact descriptions). Therefore we also adopt a similar approach as described in previous work by using random subsamples in our test [Braun 1980; Paxson 1994; Barford and Crovella 1998]. The number of samples (which are randomly picked) we use for the K-S test are 10000 throughout the paper. (In other words, we compare the computed D value with a critical value of 0.00874 in each test.)

4. TRAFFIC IS DIFFERENT ANY WHICH WAY YOU LOOK

In this section, we show that Internet traffic looks different both in time and space domain after examining the traces we obtained from different locations and at different times. These observations stress the importance of being able to parameterize models from new data to account for changes of the traffic.

4.1 Metrics Used for Comparison

We determine if two sets of data are *different* by comparing them qualitatively and quantitatively.

By qualitatively, we visually inspect the CDF plots of first-order statistics at three different levels (i.e. packet-, flow- and user-level statistics) and the wavelet scaling plots between the trace and model to see if they match closely. Here we define a *flow* as a unidirectional series of IP packets traveling between a source and a destination IP/port pair within a certain period of time, and a unique IP address as a *user*. Specifically, the metrics we use for comparison

Table II. Summary of Protocol Mix of ISI-1 Traffic

Protocol	Inbound	Outbound
NNTP (% packets)	39.4%	8%
(% bytes)	**64.4%**	0.02%
(% no. of flows)	0.06%	0.08%
HTTP (% packets)	15.8%	27.6%
(% bytes)	20.0%	**50%**
(% no. of flows)	38.5%	35.8%
DNS (% packets)	29.9%	31.6%
(% bytes)	4.8%	4.8%
(% no. of flows)	51.4%	30.1%
FTP (% packets)	5.5%	20.4%
(% bytes)	4.1%	33.7%
(% no. of flows)	8.7%	26.2%
OTHERS (% packets)	9.4%	20.4%
(% bytes)	6.7%	13.3%
(% no. of flows)	1.5%	7.8%

include packet inter-arrival time, packet size, flow duration, flow size, flow inter-arrival, user inter-arrival, user duration, protocol mix and traffic volume. We only show the CDFs of flow statistics and wavelet scaling plots in this paper for brevity since they are less dependent on the density of traffic.

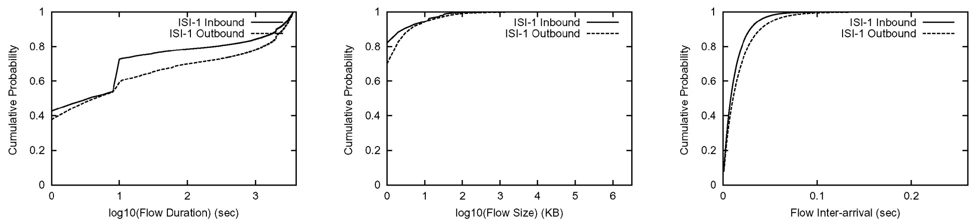
By quantitatively, we perform the Kolmogorov-Smirnov Test, as described in Section 3.3, to see if the distributions of trace and model are statistically different from each other.

4.2 Traffic Seen in Different Directions

First we look at traffic flows in different directions (i.e. inbound traffic versus outbound traffic) during the same period of time. We found inbound traffic and outbound traffic are significantly different in terms of protocol mix and via comparison of first-order statistics and wavelet analysis.

The protocol mixes for inbound and outbound traffic of ISI-1 data are shown in Table II. The traffic mix is noticeably different in different directions, where the inbound traffic is dominated by News traffic while the outbound traffic mainly consists of web and FTP traffic. Note that NNTP traffic in outbound data mainly consists of ACKs, which is the reason it contributes very little in terms of bytes to the total traffic volume. In terms of the number of flows, the majority of the flows are contributed by DNS traffic in inbound traffic while by web traffic in outbound data.

We next look at the first-order statistics. The comparison of flow statistics, including flow duration, flow size and inter-arrival time of inbound and outbound data are shown in Figure 1. Outbound traffic has comparatively longer flow duration and size than inbound traffic, which is possibly due to the fact that the majority of the flows are contributed by DNS traffic in inbound traffic while by web and FTP in outbound traffic, as shown in Table II. Although in Figure 1(b) and Figure 1(c) the CDF plots for outbound and inbound traffic look similar in the tail of the distributions (lower tail in flow size and upper tail in flow inter-arrival), none of them passes the Kolmogorov-Smirnov test.



(a) Comparison of flow duration of inbound and outbound traffic.

(b) Comparison of flow size of inbound and outbound traffic.

(c) Comparison of flow inter-arrival time of inbound and outbound traffic.

Fig. 1. Comparison of flow statistics in ISI-1 data.

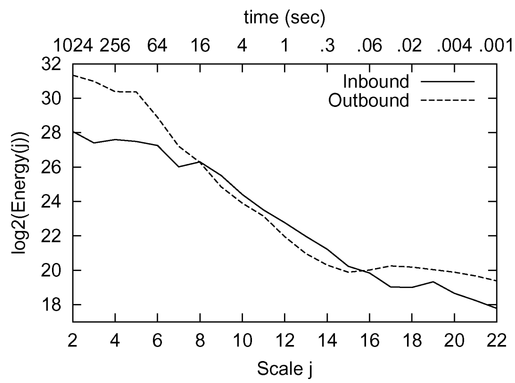


Fig. 2. Comparison of wavelet scaling plots of inbound and outbound traffic in ISI-1 data.

The D values for Figure 1(b) and Figure 1(c) are 0.121 and 0.097 respectively, which are larger than the critical value and hence fail the test. (The number of samples we use are 10000 which corresponds to a critical value of 0.00874.)

The corresponding wavelet scaling plot is shown in Figure 2. We observe there is a pronounced dip on the order of about 128ms, which reflects the underlying periodic component (i.e. RTT) for outbound traffic, while the dominant RTT for inbound traffic is on a relatively smaller time scales (about 40ms).

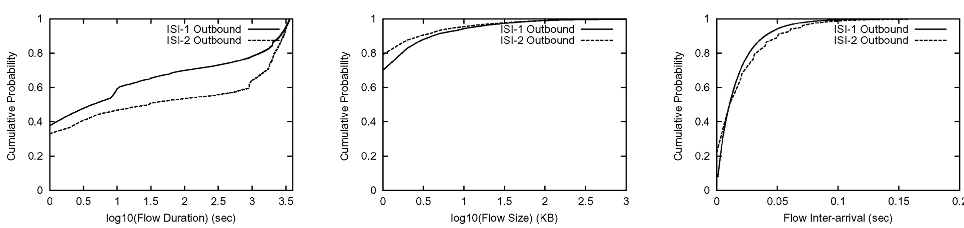
All the statistics conclude that ISI-1 inbound and outbound traffic are noticeably different from each other.

4.3 Traffic Seen at Different Times

We next look at two sets of ISI traffic captured at different times (i.e. ISI-1 and ISI-2). Here we concentrate on the comparison of outbound traffic. Since ISI-2 data was recorded during the time when most people have left the office, intuitively the inbound traffic in ISI-2 will be different from ISI-1 because of its smaller user population. (For inbound traffic, ISI-1 has 517 users while ISI-2 has only 128 users. For outbound traffic, ISI-1 has 16447 users and ISI-2 has 14259 users.) The following statistical comparisons show that ISI-1 and ISI-2 outbound traffic are different from each other.

Table III. Summary of Protocol Mix of ISI Outbound Traffic at Different Times

Protocol	ISI-1	ISI-2
NNTP (% packets)	8%	10%
(% bytes)	0.02%	0.02%
(% no. of flows)	0.08%	0.09%
HTTP (% packets)	27.6%	17.5%
(% bytes)	**50%**	24.0%
(% no. of flows)	35.8%	32.6%
DNS (% packets)	31.6%	41.0%
(% bytes)	4.8%	11.4%
(% no. of flows)	30.1%	34.5%
FTP (% packets)	20.4%	22.1%
(% bytes)	33.7%	**45.7%**
(% no. of flows)	26.2%	31.3%
OTHERS (% packets)	20.4%	9.4%
(% bytes)	13.3%	18.9%
(% no. of flows)	7.8%	7.0%



(a) Comparison of flow duration between ISI-1 and ISI-2.

(b) Comparison of flow size of between ISI-1 and ISI-2.

(c) Comparison of flow inter-arrival time between ISI-1 and ISI-2.

Fig. 3. Comparison of flow statistics for ISI outbound traffic at different times.

First we look at the traffic mix, as shown in Table III. Although large percentages of traffic in both data sets are made up by both web and FTP traffic, one is dominated by FTP while the other by web traffic.

The distributions of flow statistics including flow duration, flow size and inter-arrival time for ISI-1 and ISI-2 data are shown in Figure 3. The flow duration in ISI-2 data is significantly longer than that in ISI-1, as shown in Figure 3(a), which is probably due to the fact that ISI-1 data is dominated by web traffic while ISI-2 is dominated by FTP flows. In terms of flow size, there are more short flows in ISI-2, which is probably because there is more DNS traffic and short HTTP connections in ISI-2 data, as shown in Table III.

Again, although the CDF plots between ISI-1 and ISI-2 in Figure 3(b) and Figure 3(c) have similar shapes, they all fail the Kolmogorov-Smirnov test (the D values are 0.09 and 0.14 respectively, for 10000 samples).

The wavelet scaling plot, as depicted in Figure 4, indicates ISI-2 traffic has smaller and more heterogeneous RTT behavior shown as a dip that stretches from 8ms to 128ms while ISI-1 data has a main dip at 128ms.

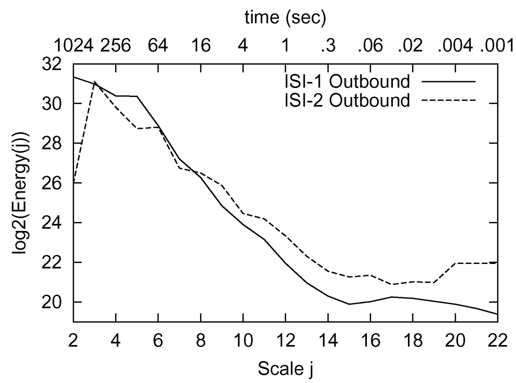
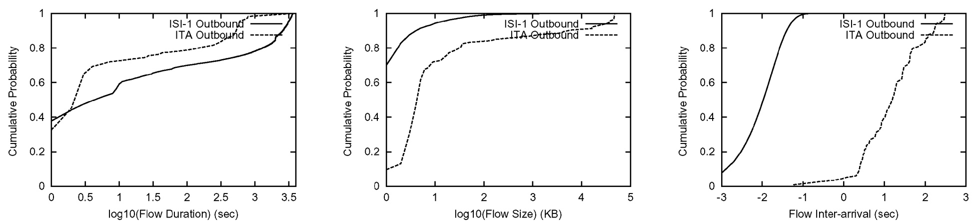


Fig. 4. Wavelet scaling plot for ISI-1 and ISI-2 outbound traffic.



(a) Comparison of flow duration between ISI-1 and ITA.

(b) Comparison of flow size between ISI-1 and ITA.

(c) Comparison of flow inter-arrival time between ISI-1 and ITA.

Fig. 5. Comparison of flow statistics for ISI and ITA outbound traffic.

All the statistical comparisons indicate that ISI-1 outbound traffic is different from ISI-2 outbound traffic.

4.4 Traffic Seen at Different Locations

Finally we look at the comparison between ISI-1 and ITA data and show traffic is different at different locations. Again, we only focus on outbound traffic.

In terms of protocol mix, ITA data only consists of HTTP and FTP traffic, which is obviously different from the protocol mix in ISI-1 traffic.

The distributions of flow statistics, including flow duration, flow size and inter-arrival time for ISI-1 and ITA data are shown in Figure 5. We see ISI-1 has longer flow duration but smaller flow size. A close look shows that the long flows in ISI-1 are mainly contributed by DNS, NTP (periodic time synchronization between servers) and NNTP traffic (periodic news exchanges between servers). ITA data has larger flow size because it mainly consists of bulk FTP transfer. It is not surprising that ITA has much larger flow inter-arrival time since its traffic is much more sparse than ISI-1. We did not apply the Kolmogorov-Smirnov Test to ITA and ISI-1 data since their CDF plots are obviously different.

In the wavelet scaling plot, as shown in Figure 6, we observe there is a main dip at a time scale of around 500ms for ITA data, which is about 4 times larger than the 128ms in ISI-1 data. A closer look shows ITA traffic is dominated by

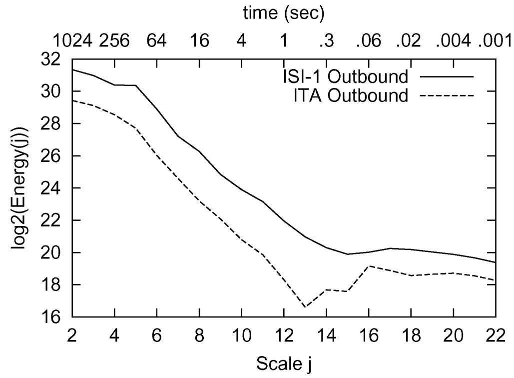


Fig. 6. Comparison of wavelet scaling plots between ISI-1 and ITA outbound traffic.

a few FTP transfers between the ITA site and some hosts in the US west coast and Europe.

All the statistical comparisons here show that traffic can be different at different sites because of the different nature of their contents.

The above discussion concludes that network traffic not only changes over time but also shows great variations in different directions and different locations. We demonstrate that the differences can be due to a variety of reasons such as user behavior, path characteristics and application usage and so on, hence it is difficult to obtain a "general" traffic model.

5. RAMP: RAPID MODEL PARAMETERIZATION

Motivated by the previous observation that it is important to quickly parameterize models from new data to account for the diversity of the traffic, we design a tool called *RAMP*. RAMP can convert live measurements into simulation models that can then be used to generate realistic synthetic traffic. In this section we describe our approaches from analyzing the trace to finally generating the simulation model.

Our approach is to automatically generate statistics that model user behaviors and network path characteristics by analyzing TCP/IP header information captured in the measurements. The resulting model will then be built into the widely-used NS network simulator [Breslau et al. 2000] and validated against the original trace via wavelet-based analysis and first order statistical comparison.

The input of RAMP is a tcpdump-format file, recorded at a single tap point of the network, that contains only TCP/IP header information. The output of RAMP is a set of CDF (Cumulative Distribution Function) files that model the corresponding traffic, as shown in Figure 7. Specifically, the CDF files consist of two types of data. One set of CDF files model user/application level statistics of the traffic, such as user session arrival, page/file size and so on. Currently RAMP only supports web and FTP traffic, which are among the most dominant types of traffic [McCreary and Claffy 2000] of the present Internet. The other one models path characteristics of the network. In particular, we focus on characterizing

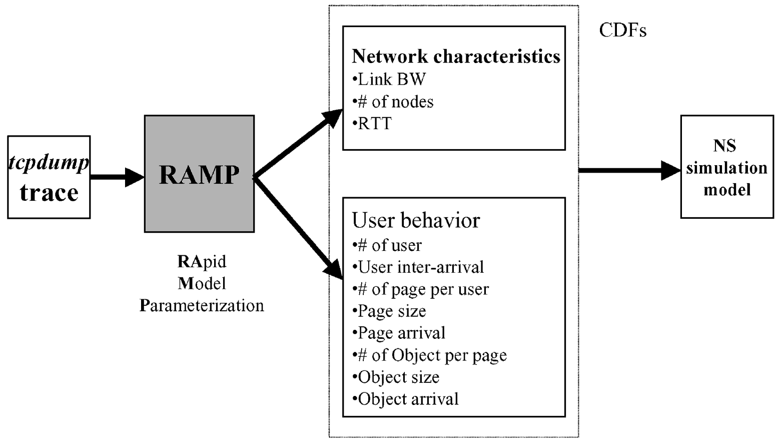


Fig. 7. Data flow of RAMP.

RTT and bottleneck bandwidth of the measured traffic since they are important parameters for driving network simulation. Typically it takes tens of minutes for RAMP to process a trace file with the size of several hundred megabytes.

5.1 User and Application Behavior Characterization

In this section we describe the techniques we employ to characterize the source-level behaviors based on the TCP/IP header information captured in the trace. We focus on the analysis of web and FTP traffic, which are among the most dominant types of traffic of the present Internet.

5.1.1 *Web Traffic.* Here we present the methodology used to characterize the important components of web traffic based on only the information in the TCP/IP headers and knowledge of the TCP and HTTP protocol.

To reconstruct the data exchanges in the HTTP connections based on only the information in the TCP/IP header, we adopt an approach and heuristics similar to previous work [Smith et al. 2001]. One observation in their study is that when the server receives a HTTP request it will send TCP acknowledgments (ACKs) indicating the in-order byte sequence it has received, and all of the request messages will be ACKed before the corresponding HTTP response data is sent (note that here we assume there is no pipelining in use). Hence we can infer the size of request by the amount of ACK value advances and the size of response by the amount of data sequence number advances. As the example shown in Figure 8, the ACK-only segment from the server following the SYN+ACK segment indicates the first request was 325 bytes in size. In the following segments, the data sequence numbers advance to 2458 (the size of first response) with no further changes in the ACK values. In the next segment, the advance of ACK number indicates the size of the second request was 349 bytes (675−326). In the following segments, the data sequence numbers advance with no further changes in the ACK values. The size of second response is 11756 bytes (14124−2458).

```
26350.296819 10.1.7.14.80 > 10.3.162.34.4645: S 247361:247361(0) ack 534867 win 8760
26350.312486 10.1.7.14.80 > 10.3.162.34.4645: . ack 326 win 8760
26350.313099 10.1.7.14.80 > 10.3.162.34.4645: P 1:1461(1460) ack 326 win 8760
26350.430730 10.1.7.14.80 > 10.3.162.34.4645: P 1461:2458(997) ack 326 win 8760
26367.549809 10.1.7.14.80 > 10.3.162.34.4645: . 2458:3918(1460) ack 675 win 8760
26367.549942 10.1.7.14.80 > 10.3.162.34.4645: P 3918:5378(1460) ack 675 win 8760
26367.550065 10.1.7.14.80 > 10.3.162.34.4645: P 5378:6838(1460) ack 675 win 8760
26367.565980 10.1.7.14.80 > 10.3.162.34.4645: . 6838:8298(1460) ack 675 win 8760
26367.566105 10.1.7.14.80 > 10.3.162.34.4645: . 8298:9758(1460) ack 675 win 8760
26367.566228 10.1.7.14.80 > 10.3.162.34.4645: P 9758:11218(1460) ack 675 win 8760
26367.581947 10.1.7.14.80 > 10.3.162.34.4645: . 11218:12678(1460) ack 675 win 8760
26367.582068 10.1.7.14.80 > 10.3.162.34.4645: P 12678:14124(1446) ack 675 win 8760
26397.549684 10.1.7.14.80 > 10.3.162.34.4645: F 14124:14124(0) ack 675 win 8760
```

Fig. 8. *tcpdump* trace that shows two request/response exchanges in a persistent HTTP connection.

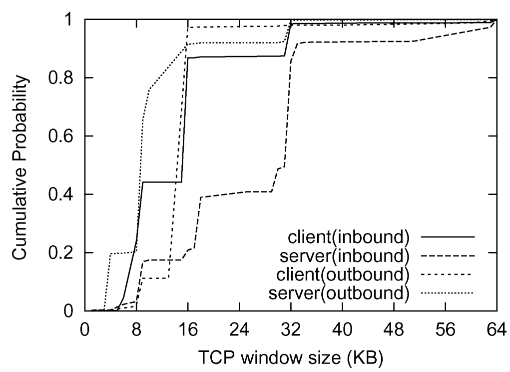


Fig. 9. Comparison of the usage of TCP window size in inbound and outbound traffic of ISI-1 data.

Adopting similar heuristics as those developed originally by Mah [1997] and Barford and Crovella [1998], we assume a new page is requested after some period of idle time (or "think" time) at the client. We identify idle periods in which either the client has no established TCP connection or no established connection has an active request/response exchange in progress.

Our web traffic model is similar to those developed originally by Mah and Barford and Crovella. However, we found it is important, but not captured by the previous studies, to model the TCP window size and the usage of persistent connection.

It is important to model TCP window size in order to accurately characterize the sending rate of the servers. For example, as shown in Figure 9, more than 80% of clients in the ISI1 inbound traffic use window size less than 16K. Using small window size will limit the servers from fully utilizing increasingly-popular broadband networks such as DSL and cable modem. Note that we did not observe any connection that uses TCP window scale option in our traces.

Motivated by the increasingly important role of persistent connection in web traffic, as reported by a previous study [Smith et al. 2001], we also model the persistent connection used in HTTP/1.1, As shown in Table IV, although only less than 20% of connections are persistent, they account for about 50% of all

Table IV. Summary for the Usage of HTTP Persistent Connections in ISI-1 Traffic

Protocol	Inbound	Outbound
Number of connections	26426	4425
objects	44399	7187
bytes	318.7 MB	424 MB
Persistent connections	4756 (18%)	708 (16%)
Objects on Persistent	22841 (51%)	3506 (49%)
Bytes on Persistent	121.5 MB (38%)	85.4 MB (20%)

objects transferred and more than 20% of all bytes transferred. This clearly shows persistent connection plays an important role in the dynamics of TCP connections for the Web. In our datasets, over 50% of persistent connections are used for three or more request/response exchanges and 10% of them carry more than nine (the graphs are not shown here). Our result for the usage of persistent connections shows strong agreement with recent studies [Smith et al. 2001]. Note that although we have observed in our datasets that some browsers still use multiple concurrent connections to transmit one single page as reported in Balakrishnan's study [Balakrishnan et al. 1998], we did not model that since it accounts for less than two percent of the total number of pages in our traces.

5.1.2 *FTP Traffic.* In this section we show that it is non-trivial to extract FTP flows in the traces. (In particular, it is not sufficient that one only looks at the flows that origin from or destine to port 20 or 21.)

For FTP traffic, we assume that a unique IP address represents a single human user and a new TCP connection is used for each file transmission. This heuristic allows us to identify the points when the client starts a new file. The FTP protocol [Postel and Reynolds 1985] specifies that the client first connects from a random unprivileged port ($N > 1024$) to the FTP server's command port, port 21. The client then starts listening to port N+1 and sends the FTP command "PORT N+1" to the FTP server. The server will then connect back to the client's specified data port from its local data port which is port 20. This is also known as Active-mode FTP.

However, from our datasets we observed that there are a significant number of clients using Passive-mode FTP, in which the client initiates both control and data connections to the server. When opening an FTP connection, the client opens two random unprivileged ports locally ($N > 1024$ and N+1). The first port contacts the server on port 21, but instead of then issuing a PORT command and allowing the server to connect back to its data port, the client will issue the PASV command. The result of this is that the server then opens a random unprivileged port ($P > 1024$) and sends the PORT P command back to the client. The client then initiates the connection from port N+1 to port P on the server to transfer data. To identify FTP traffic, we first locate FTP clients by looking at those connected to server port 20 and find out what are the control ports (N) they use. We then look for the connections originating from the neighboring ports (N+1) of client's control port and classify them as FTP data connections.

5.2 Characterization of Network Path Properties

In this section we describe how we estimate the topology information from the measurement. Particularly we focus on characterizing the round trip delay and bottleneck bandwidth since both of them are important for driving the simulation.

5.2.1 *Round-trip Delay.* We determine the RTT of each TCP connection in our traces by computing the difference of timestamp between data packet and the first ACK packet which has the same sequence number. However, this approach is not applicable for packets captured at the data receivers end, where the timestamp difference between data and ACK doesn't reflect the path delay. For situations where the clients are near the measurement point while servers are at the remote end (eg. the inbound traffic), we rely on the three-way handshake at the start of each TCP connection to calculate the delay of the path. In other words, we compute the RTT by taking the timestamp difference between the SYN packet and its corresponding ACK. For each connection we take the minimum of RTT samples as an approximation of propagation delay of the path (after dividing the RTT by 2) and consider the deviations from the minimum RTT as variances caused by queuing delay and transmission delay. We use this approximation to drive our simulation.

5.2.2 *Bottleneck Bandwidth.* Our traces contain both outbound and inbound traffic. For outbound traffic, we use Sender Based Packet Pair (SBPP) [Paxson 1997] to compute the bottleneck bandwidth between the local servers and the remote clients. That is, we estimate the spacing between a pair of back-to-back TCP packets after passing the bottleneck link by examining the arrival times of their corresponding ACKs (for delayed-ACK packets, we estimate the spacing between the second and the forth packets of a group of 4 back-to-back packets). For inbound traffic, we rely on Receiver Only Packet Pair (ROPP) [Lai and Baker 1999], which uses the arrival times of two consecutive full-size packets at the receiver to estimate the bottleneck bandwidth between remote servers and the local clients. We also apply similar techniques to filter noise such as density estimation as described in Lai and Baker [2001].

5.3 Structural Simulation Model

Traditional black box approaches typically treat the measurement as a time series. They focus on capturing the statistical characteristics (particularly autocorrelation and marginal distribution) of empirical data to model network traffic, based on various approaches such as Markov process, ARIMA, TES etc. [Heffes and Lucantoni 1986; Maglaris et al. 1988; Sen et al. 1989; Nikolaidis and Akyildiz 1992; Lucantoni et al. 1994; Heyman et al. 1994; Gurenefelder et al. 1991; Melamed and Sengupta 1992]. Although they are able to reproduce the measured traffic correctly, these approaches generally ignore the underlying network structure and hence provide little or no insight about the observed characteristics of measured traffic and its underlying causes. On the other hand, structural modeling, first discussed by Willinger [Willinger et al. 1998], proposes that we should implicitly take into account the complex

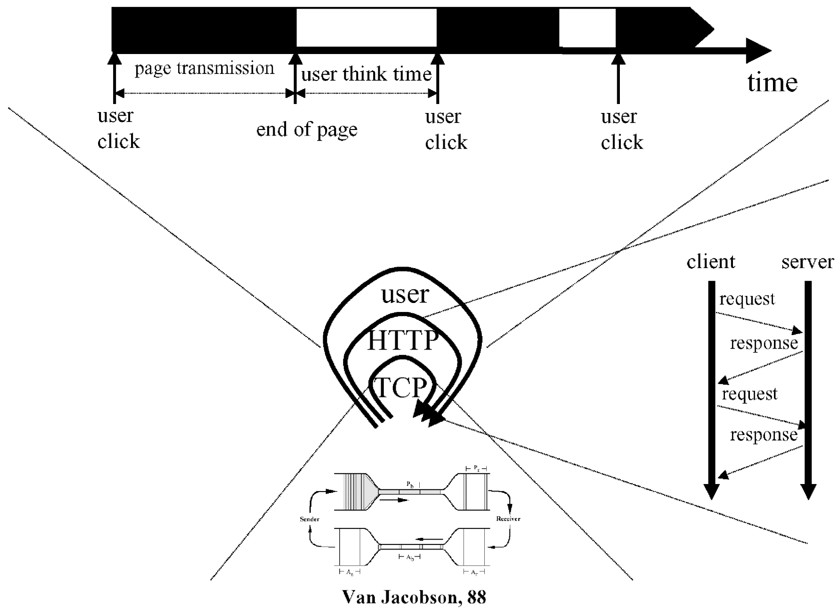


Fig. 10. Multiple levels of feedback in web traffic.

hierarchical structure of application and intertwined networking mechanisms in order to accurately reproduce the traffic while still providing a physical explanation for observed phenomena.

As opposed to trace-replay, there are several advantages to this approach:

— Some protocols must be modeled as end-to-end entities in order to capture the feedback effect such as TCP congestion control, while trace-replay techniques typically ignore the fact that traffic is frequently *shaped* by the network's current properties,
— Internet protocols present very rich, multi-fractal behavior across a range of time scales. Simple trace-replay approach will fail to capture this richness.
— By capturing the details of data transfer in an algorithm we can reproduce that traffic with much less storage requirements than trace-replay.

As shown in Figure 10, we can see that there are multiple levels of feedback effect within the hierarchical structure of web traffic, and each level operates at different time scales. For example, TCP has its own congestion control mechanism, which operates at the time scale of seconds, while HTTP has the request-response loop functioning at the time scale of tens of seconds. Hence, it is important to reproduce the structure of application in the model in order to accurately reproduce the traffic.

Based on the structural modeling approach, we design a three-level simulation model to characterize web traffic and two-level model to characterize FTP traffic as shown in Figures 11 and 12. Note that we only model the data connections of FTP traffic for simplicity since the bandwidth usage of the control channel is negligible (typically less than one percent of total traffic in our datasets).

User behavior

(1) User arrival is modeled as a Poisson process with certain rate.
(2) The number of pages per user session is randomly picked from the CDF(Cumulative Distribution Function) of trace.
(3) the source of page are chosen from a CDF that matches the popularity of servers
(4) Each page is sequentially requested by the users as described below.

Page

(1) Page size is chosen from a CDF
(2) The inter-arrival time of page is picked from a CDF
(3) The number of objects within one page is picked from a CDF
(4) The size of request to a page is picked from a CDF
(5) User decides a TCP connection is used for multiple request/response exchanges or a single request/response exchange based on the probability of persistent connection (HTTP1.1) versus non-persistent connection (HTTP1.0) computed from the trace. In persistent connection mode, all objects within the same page are sent via the same TCP connection.

Object

(1) The inter-arrival time of object is picked from a CDF
(2) The size of object is picked from a CDF
(3) The TCP window size for both servers and clients are also randomly chosen from a CDF

Fig. 11. Structural model of web traffic.

User behavior

(1) User arrival is modeled as a Poisson process with certain rate.
(2) The number of file transmitted per user session is randomly picked from the CDF(Cumulative Distribution Function) of trace.
(3) the source of file are chosen from a CDF that matches the popularity of servers
(4) User starts a new TCP connection for each new file which is sequentially transmitted as described below.

File

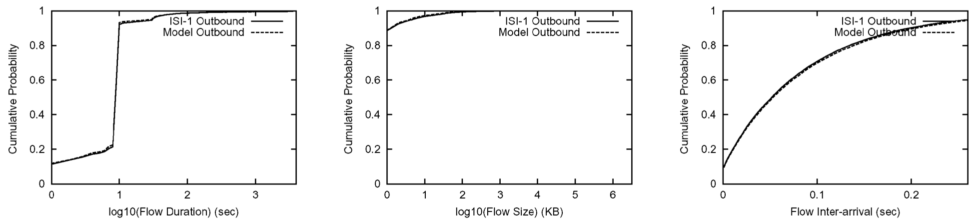
(1) file size is chosen from a CDF
(2) The inter-arrival time of file is picked from a CDF
(3) The TCP window size for both servers and clients are also randomly chosen from a CDF

Fig. 12. Structural model of FTP traffic.

Our web traffic model is similar to those developed originally by Barford and Crovella [1998]. Additionally, we model TCP window size and the usage of persistent connection. We also model HTTP request size motivated by the trend in using large requests due to the increasing popularity of "web email" [Smith et al. 2001].

6. EVALUATION OF RAMP

To validate if RAMP accurately reproduces the traffic under study, we incorporate its output into an ns-2 simulator and compare the result of simulation against the original traces. To understand if RAMP can perform as well as existing work in terms of generating realistic synthetic workload, we also compare RAMP with SURGE [Barford and Crovella 1998], a popular web traffic workload generator. Another important aspect is to understand if RAMP is



(a) Comparison of flow duration between model and ISI-1 trace.

(b) Comparison of flow size between model and ISI-1 trace.

(c) Comparison of flow inter-arrival time between model and ISI-trace.

Fig. 13. Comparison of flow statistics for model and ISI-1 outbound traffic.

parsimonious. A strength of mathematical models is their parsimony — with only a few parameters they are able to describe same class of traffic. To evaluate RAMP's complexity, we look at it in two different aspects: first, the effectiveness of using an analytical model versus an empirical model; second, the importance of details in its structural model.

6.1 Comparison with Original Traces

In this section we use ISI-1 data to evaluate the accuracy of RAMP. The result shows that the output of the simulation matches the original traces closely. Note that because currently our tool only supports web and FTP traffic, we first filter the traces so that they only contain web and FTP data before being compared against the simulation result (together web and FTP traffic account for 83.7% of the total traffic in terms of the number of bytes, and 48% in terms of the number of packets in ISI-1 trace).

The statistics we use here for validation including the distributions of flow arrival, flow size, flow duration, packet inter-arrival time, wavelet scaling plot and the application-specific parameters, such as page size, page arrival, object size (for web traffic), file size, file arrival (for FTP traffic), user arrival and user duration. Again, here we only show outbound traffic and only CDF plots of flow statistics for simplicity (although the graphs of inbound traffic are not shown here, they are consistent with the results of outbound traffic).

The CDF plots of flow statistics for ISI-1 model are depicted in Figure 13, which shows that the model matches the trace closely. The Kolmogorov-Smirnov test D values for Figure 13(a), Figure 13(b) and Figure 13(c) are 0.0019, 0.0013, 0.0018 respectively. They all pass the K-S test given a critical value of 0.00874.

The corresponding wavelet scaling plot for the ISI-1 model, as depicted in Figure 14, also shows a large degree of resemblance between trace and model, such as similar energy value (the model has slightly lower energy though) and a dip around 128ms (which reflects the RTT of the underlying traffic).

The CDF plots of model parameters such as page/file size, user arrival etc. also match closely (not shown here), which is not surprising since the model is directly driven by those parameters.

All the statistical comparisons show RAMP is able to accurately reproduce the original traffic.

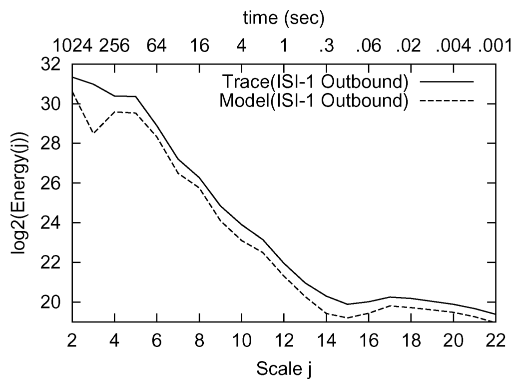


Fig. 14. Comparison of wavelet scaling plots between model and trace for ISI-1 outbound traffic.

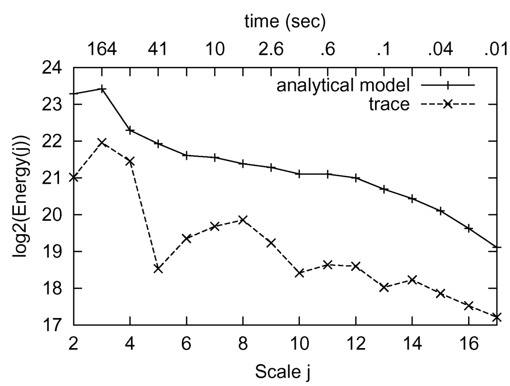


Fig. 15. Comparison of wavelet scaling plots between SURGE-like analytic model and a trace for ITA web traffic.

6.2 Comparison with Existing Tools

In order to understand if RAMP can generate representative workload, we compare RAMP against an existing traffic generator, namely SURGE [Barford and Crovella 1998]. We demonstrate that our model parameterization tool is capable of achieving the same functionality of SURGE (i.e. generating similar traffic workload like SURGE) without suffering its limitation due to some of its implicit assumptions.

SURGE contains a set of programs that pre-compute several datasets and a multi-threaded program that makes web requests using those datasets. Both are written in C. The datasets consist of the distribution models of a number of requests, file sizes, popularity of files, embedded objects, file temporal locality and OFF time.

To validate RAMP against SURGE, we performed a lab experiment by running SURGE for 30 minutes and recording the traffic via *tcpdump*. We then fed the SURGE trace into RAMP and inspected whether the output of ns-2 simulation model from RAMP agreed with the SURGE trace. The environment of the

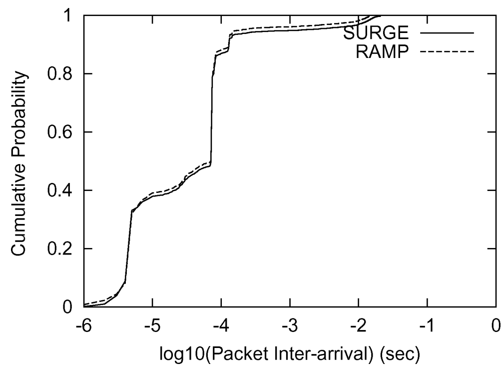


Fig. 16. Comparison of packet inter-arrival time between SURGE and RAMP.

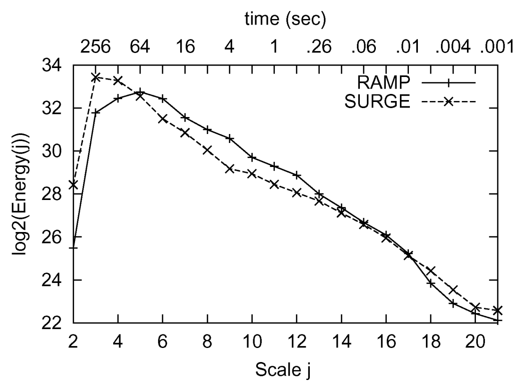


Fig. 17. Comparison of wavelet scaling plots between SURGE and RAMP.

experiment consists of five PCs connected by an 10MBps Ethernet switch. Four of these boxes are used as SURGE clients that are Pentium II/III class (266MHz and above) Linux boxes. We used a Pentium IV Linux box (1.7GHz with 750M memory) as the SURGE server, which ran Apache v.1.3.22. The number of UE (user entity, SURGE's representation of a web user) and CP (client process, which decides how the threads are spawned) are 5 and 50 respectively. We ran SURGE v.1.00a with HTTP 1.0.

We look at the packet inter-arrival time and wavelet scaling plot of the outputs of SURGE and our model respectively. All the statistics match closely, as shown in Figures 16 and 17.

One limitation of SURGE is that it attempts to fit the models into some widely-used analytic functions (such as using Pareto to describe the distributions of file sizes and off time). However, it is not universally true that all of the web traffic follows these assumptions. For example, these assumptions might break for a trace distribution site like ITA. We have observed that the distribution of page size in ITA traffic (which are mainly made up by plain HTML files that describe traces and collection/analysis software) is not heavy-tailed, and hence can not be modeled by SURGE. The presence of heavy tails typically is indicated by an approximately straight line in the tail in the LLCD plot

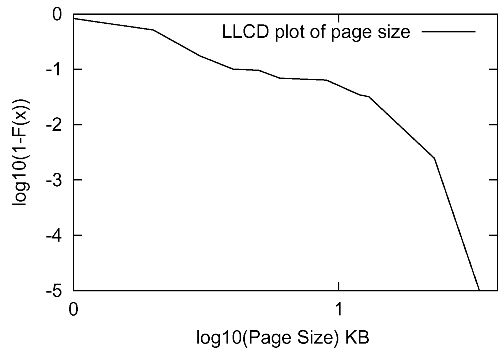


Fig. 18. Log-Log Complementary Distribution plot of Page Size in ITA traffic.

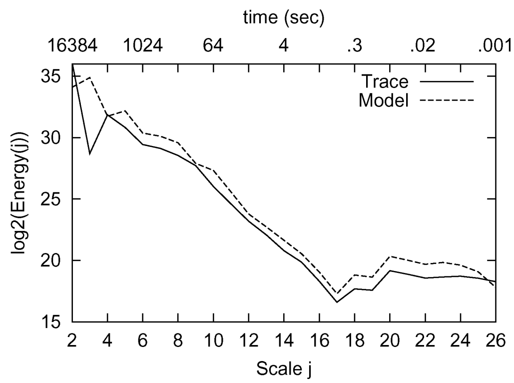


Fig. 19. Comparison of wavelet scaling plots between model and trace for ITA outbound traffic.

[Crovella and Bestavros 1996], which we do not observe in ITA data, as shown in Figure 18.

6.3 Analytical Model vs. Empirical Model

Due to the diversity of the Internet traffic, for some websites an analytic model is not sufficient to capture the traffic characteristics, To demonstrate this aspect, we simulate the web traffic in ITA data with a SURGE-like analytic model in ns-2, similar to models used in previous work [Feldmann et al. 1999; Huang et al. 2001]. We show that this SURGE-like workload model does not accurately reproduce the ITA web traffic. As the wavelet plots shown in Figure 15, the traffic generated by the analytic model does not capture the scaling features of ITA traffic at both small and large time scales.

On the other hand, our tool is based on empirical distributions of traffic and does not have any implicit assumptions about the distribution of the traffic, hence it is more flexible to cope with the diversity of the traffic. According to the wavelet plot shown in Figure 19, the ITA model generated by RAMP does capture the important features of ITA traffic (such as a dip at 500ms and similar energy levels).

User behavior

(1) User arrival is modeled as a Poisson process with a certain rate.
(2) The number of flows per user session is randomly picked from the CDF of the trace.
(3) the source of flows are chosen from a CDF that matches the popularity of the servers
(4) Each flow is sequentially requested by the users as described below.

Flow

(1) Flow size is chosen from a CDF
(2) The inter-arrival time of flow is picked from a CDF

Fig. 20. Two level flow-based model of web traffic.

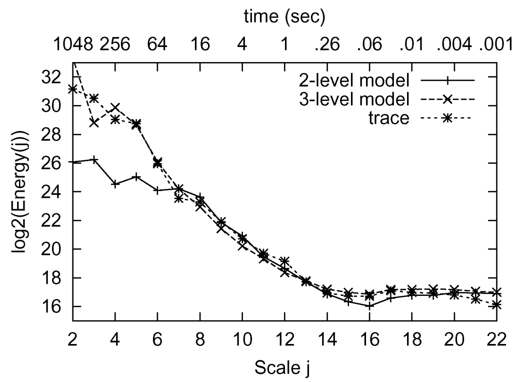


Fig. 21. Comparison of wavelet scaling plots between 2-level and 3-level web models.

6.4 Effect of Detail in a Structured Model

In our study we develop a three-level model, as shown in Figure 11, to capture the characteristics of web traffic. To understand the importance of capturing the details of application-level structure in order to correctly model the traffic, we compare the results of using a simplified flow-based model (where the hierarchical relationship between page and object has been omitted, as shown in Figure 20) against our original web model. As shown in Figure 21, although the plots look similar at smaller time scales, the traffic generated by the simplified two-level web traffic model becomes less bursty at larger time scales (larger than 16 seconds). Hence, we conclude that it is important to capture the details of application-level structure (a three-level model rather than a two-level model in this case) in order to accurately reproduce the traffic. This example also shows that use of empirical distributions from real traces provides no guarantee of model accuracy; differences in application structure also have an important affect on simulation accuracy.

7. PERFORMANCE OF RAMP

The time required for RAMP from analyzing the traces to finally generating the simulation models typically takes tens of minutes for a trace with size of several hundred megabytes, although the process speed also depends on the nature of the traffic (currently RAMP only supports web and FTP traffic) and its actual volume. In this section, we show the speed of RAMP is a function of

Table V. Process Time of RAMP for Different Traces

Trace	ISI-1	ISI-2	ITA
file size (MB)	614	561	203
no. of bytes (GB)	1.0	7.3	2.4
no. of packets (M)	9.2	8.4	2.5
no. of flows (K)	506	398	1.3
process time (min)	25	21	8
speed (thousand packets/sec)	6.1	6.3	5.7

the number of packets in the trace file. Currently we support traces captured in tcpdump format.

To understand what are the factors that will affect the performance of RAMP, we ran RAMP on a 1.7G Hz Pentium IV Linux box with 1GB memory for different trace files obtained at different times and different places. As shown in Table V, we can see the process time of RAMP is approximately proportional to the number of packets in the trace (and hence also proportional to the file size). In general, it takes tens of minutes for RAMP to process an hour-long trace, allowing users to simulate *current* traffic several times per day.

8. LIMITATION

In this section, we describe some inherent limitations that will affect our results. These limitations include the uncertainties when reconstructing HTTP level information from TCP/IP header, incomplete flows in the traces and the limitation of estimating bandwidth based on passive measurement.

Our methodology to infer the source behavior of web traffic is based on the limited information available in TCP/IP header for one direction of a TCP connection. There are a number of uncertainties arising from issues such as pipelining, user/browser behavior, caches and TCP segment re-ordering that will affect our inference, as described by Smith et al. [2001]. However, we expect these cases will typically only appear as a very small percentage of total traffic and will not noticeably affect the normal operating condition of our model parameterization tool.

We find incomplete TCP connections at the beginnings and ends in the data since our traces only cover specific intervals of time (i.e. one hour). We excluded these incomplete connections from our analysis. However, we expect this might have some effect on the results since it will affect some of the model parameters (eg. page size and number of objects per page). In our study the incomplete connections account for 2-4% of the total connections. Since we ignore these connections, we expect that our model will underestimate traffic volume. To quantify this error we analyzed the distribution of long flows in two 24-hour long traces from NLANR [2001]. Although there are a small number of flows longer than an hour (0.02% by flow count, about 5% by packet count) if we examine all flows of the NLANR traces, the majority of these flows are NNTP traffic. Examining merely web and FTP traffic we see only 0.006% of flows or 0.01% of packets are in flows longer than an hour. Therefore we believe that our model will not significantly underestimate web and FTP traffic.

There are some known issues with using SBPP and ROPP to measure the bottleneck bandwidth. For example, cross traffic and post-bottleneck queuing tend to distort the estimation. Previous study by Lai et al. showed that the inaccuracy of bandwidth estimation based on passive measurements can be as high as 41% [Lai and Baker 2001]. However, as shown in Section 6, our results indicate that these techniques combined with some simple filtering mechanism give us reasonable approximation to estimate bottleneck bandwidth for driving our simulation model.

9. FUTURE WORK

For future work, we describe some possible improvements to RAMP. These improvements include a better queuing model, support of backbone-style traffic, real-time model parameterization, support of other types of important traffic, further validation of RAMP with traces having different characteristics, modeling of temporal relationships among different types of traffic, long-term traffic prediction, and integration of distributed measurements.

We model queuing delay as an extra component of propagation delay instead of the end result of interaction between aggregation of flows and limited buffer size (which is hard to characterize just by looking at TCP/IP header information). This approach is sufficient for our data sets, which have low link utilization and zero packet drop. However, for sites that experience serious congestion (like flash crowd), our approximation might introduce some inaccuracy in the result and require further study.

The current design of RAMP has an implicit assumption that the measured traffic is captured at the edge link (such as the link between a campus network and its ISP), so that the end-to-end path characteristics such as bottleneck bandwidth can be estimated via passive measurements. When applying RAMP to backbone-style traffic, we expect this limitation can be ameliorated with extra information obtained using existing active probing techniques [Jacobson 1997; Downey 1999; Carter and Crovella 1996].

Currently RAMP takes a trace file as input and processes the traffic off-line. Although for our current processing power and trace traffic, RAMP processing is slightly slower than real time, with slightly more computing power (or slightly lower-speed traces) and minor software changes, RAMP could parameterize the model in real-time. The primary change to RAMP would be to incrementally update the output CDFs as each new flow arrives, instead of computing all flows at once.

Our tool currently supports web and FTP traffic, which only accounts for a subset of real network traffic. To make the output of RAMP more representative, we would like to incorporate other types of important traffic such as DNS, multimedia traffic (such as Real Audio/Video) and increasingly popular peer-to-peer traffic (such as Morpheus) into our tool.

In this study, we use only two set of traces (from ISI and ITA respectively) for the design and validation of RAMP. We plan to collect more traces from other places, particularly those that potentially have very different traffic characteristics (such as at a very high speed link or a very congested site), to further investigate and validate RAMP.

To accurately model traffic, it is important to characterize the temporal relationship between different types of traffic. For example, DNS behavior is very likely linked closely to web traffic patterns since most of the web connections are preceded by DNS lookups. We plan to study this issue and understand how to correctly orchestrate different traffic classes in the model.

Currently our model is based on the trace recorded at a single tap point of the network. However, distributed measurement is required in order to get a network-wide view of traffic and correctly model the behavior of cross traffic, while keeping the size of collected data maintainable. To integrate distributed data will require approaches for overlap detection and hole filling. To address this problem, we plan to explore and extend the techniques developed in previous work of distributed network monitoring such as SCAN [Govindan et al. 1997] and recent work in network tomography [CAIDA 2002; Mathis and Mahadavi 1996; Vardi 1996; Cao et al. 2000a, 2000b], and employ new algorithms and tools to merge distributed data into a coherent model.

Measurement study of Internet traces shows that the WAN performance is reasonably stable over terms of several minutes; meanwhile, nearby hosts experience similar or identical throughput performance within a time period measured in minutes [Balakrishnan et al. 1997; Paxson 1997]. Our model parameterization tool outputs simulation model at the time scale of tens of minutes for hour-long traffic, which matches the level of stability reported in previous study and hence is applicable to simulate *present* traffic and predict the short-term traffic trend. However, to simulate and predict the long-term trend of traffic (for example, at the time scale of days), we need to understand how the traffic evolves and correlates in time.

10. CONCLUSION

Floyd and Paxson [2001] characterized the problems—the constantly-changing and decentralized nature of the Internet, resulting in a poor understanding of traffic characteristics and making it difficult to define a typical configuration for simulating it. Motivated by their observations, we developed a tool called *RAMP* that supports rapid parameterization of live network traffic for generating realistic application-level simulation models. Our model is based on estimation of user behaviors and network conditions from captured tcpdump traces. We validate our methodology by comparing some first order statistics of traces against the simulation output of the model. We also apply multi-scaling analytic techniques to debug and validate the model. In this paper, we first demonstrate that traffic is different in both temporal and spatial space. We then show the effectiveness of our approaches in terms of the capability of generating simulation models that capture traffic dynamics in a timely fashion even when facing the ubiquitous heterogeneity of the Internet. Our work has three primary results. First, we strengthen Floyd and Paxson's arguments by showing that network characteristics not only change over time but also vary in other dimensions such as locations and flow directions. Second, we propose a methodology for rapidly parameterizing traffic models. This approach employs a trace-analysis tool that infers traffic and topology characteristics, and a CDF-based traffic model that can capture widely varying web traffic. Finally, we show how our models can be automatically and rapidly parameterized from traces, allowing a user to quickly instantiate models that represent current, local traffic.

ACKNOWLEDGMENTS

We would like to acknowledge the many contributions of Polly Huang to this work. Her contributions include software for producing global scaling plots and more importantly some technical conversations about how to interpret the output of wavelet scaling analysis. We would also like to thank Walter Willinger for providing useful insight in interpreting the results of the Kolmogorov-Smirnov test. For collection of traces we would like to thank Vern Paxson of ICIR and Jim Koda of ISI for their help. We would like to thank F. Donelson Smith and Felix Hernandez Campos for providing part of their code to us for reconstructing HTTP connection information [Smith et al. 2001]. We would also like to thank Purushotham Kamath for providing helpful feedback from using RAMP in his work [Komath et al. 2002], and Tim Buchheim for helping integrate RAMP with other research activities.

REFERENCES

ABRY, P. AND VEITCH, D. 1998. Wavelet analysis of long-range-dependent traffic. *IEEE Trans. Inform. Theory 44*, 1, 2–15.

ASABA, T., CLAFFY, K., NAKAMURA, O., AND MURAI, J. 1992. An analysis of international academic research network traffic between japan and other nations. In *Inet '92*. 431–440.

BALAKRISHNAN, H., PADMANABHAN, V. N., AND KATZ, R. H. 1998. Tcp behavior of a busy internet server: Analysis and improvements. In *Proceedings of the IEEE Infocom*. IEEE, San Francisco, CA, USA.

BALAKRISHNAN, H., STEMM, M., SESHAN, S., AND KATZ, R. H. 1997. Analyzing stability in wide-area network performance. In *SIGMETRICS/Performance*.

BARFORD, P. AND CROVELLA, M. 1998. Generating representative web workloads for network and server performance evaluation. In *Proceedings of the ACM SIGMETRICS*. ACM, Madison WI, 151–160.

BRAUN, H. 1980. A simple method for testing goodness of fit in the presence of nuisance parameters. *J. Royal Statis. Society. Series B (Methodological) 42*, 1, 53–63.

BRESLAU, L., ESTRIN, D., FALL, K., FLOYD, S., HEIDEMANN, J., HELMY, A., HUANG, P., MCCANNE, S., VARADHAN, K., XU, Y., AND YU, H. 2000. Advances in network simulation. *IEEE Computer 33*, 5 (May), 59–67. Expanded version available as USC TR 99-702b at http://www.isi.edu/ johnh/PAPERS/Bajaj99a.html.

CAIDA. 2002. Internet measurement infrastructure. http://www.caida.org/analysis/performance/measinfra/.

CAO, J., CLEVELAND, W. S., LIN, D., AND SUN, D. X. 2001. On the nonstationarity of internet traffic. In *SIGMETRICS/Performance*. 102–112.

CAO, J., DAVIS, D., WIEL, S., AND YU, B. 2000a. Time-varying network tomography: Router link data. *J. Amer. Stat. Assoc. 95*, 452 (Feb.), 1063–1075.

CAO, J., WIEL, S. V., YU, B., AND ZHU, Z. 2000b. A scalable method for estimating network traffic matrices. *Bell Labs Tech. Rep.*.

CARTER, R. AND CROVELLA, M. 1996. Measuring bottleneck link speed in packet-switched networks. In *In PERFORMANCE '96, the International Conference on Performance Theory, Measurement and Evaluation of Computer and Communication Systems*.

CHANDRA, M., SINGPURWALLA, N. D., AND STEPHENS, M. A. 1981. Kolmogorov statistics for tests of fit for the extreme-value and weibull distributions. *J. Amer. Stat. Assoc. 76*, 375 (Sept.), 729–731.

CHESHIRE, S. AND BAKER, M. 1995. Experiences with a wireless network in MosquitoNet. In *Proceedings of the IEEE Hot Interconnects Symposium '95*.

CROVELLA, M. E. AND BESTAVROS, A. 1996. Self-similarity in world wide web traffic: evidence and possible causes. In *Proceedings of the ACM SIGMETRICS*. ACM, Philadelphia, Pennsylvania, 160–169.

DOWNEY, A. B. 1999. Using Pathchar to estimate internet link characteristics. In *SIGCOMM*. 222–223.

HEYMAN, D., TABATABI, A., LAKSBMAN, T., AND HEEKE, H. 1994. Modeling teleconference traffic from vbr video coders. *Proceedings ICC, IEEE*, 1744–1748.

FELDMANN, A. 2000. BLT: Bi-layer tracing of HTTP and TCP/IP. *WWW9 / Computer Networks 33*, 1–6 (May), 321–335.

FELDMANN, A., GILBERT, A. C., HUANG, P., AND WILLINGER, W. 1999. Dynamics of IP traffic: A study of the role of variability and the impact of control. In *Proceedings of the ACM SIGCOMM*. ACM, Cambridge, MA, USA, 301–313.

FLOYD, S. AND PAXSON, V. 2001. Difficulties in simulating the Internet. *ACM/IEEE Trans. Networking 9*, 4 (Feb.), 392–403.

GOVINDAN, R., ALAETTINOĞLU, C., AND ESTRIN, D. 1997. Self-configuring active network monitoring (SCAN). http://www.isi.edu/scon/Pubs/scan_proposal.ps.gz.

GURENEFELDER, R., COSMAS, J. P., MANTHROPE, S., AND ODINMA-OKAFOR, A. 1991. Characterization of video codecs as autoregressive moving average processes and related queueing system performance. *IEEE J. Selected Areas Comm. 9*, 284–293.

HEFFES, H. AND LUCANTONI, D. M. 1986. A markov modulated characterization of packetized voice and data traffic and related statistical multiplexer performance. *IEEE J. Selected Areas Communications 4*, 856–868.

HUANG, P., FELDMANN, A., AND WILLINGER, W. 2001. A non-intrusive, wavelet-based approach to detecting network performance problems. In *Proceedings of ACM SIGCOMM Internet Measurement Workshop 2001*. San Francisco Bay Area.

JACOBSON, V. 1997. Pathchar, April 1997 MSRI talk.

KAMATH, P., LAN, K., HEIDEMANN, J., BANNISTER, J., AND TOUCH, J. 2002. Generation of high bandwidth network traffic traces. In *Proceedings of MASCOTS '02*. IEEE, Fort Worth, Texas, USA, 401–410.

LAI, K. AND BAKER, M. 1999. Measuring bandwidth. In *INFOCOM (1)*. 235–245.

LAI, K. AND BAKER, M. 2000. Measuring link bandwidths using a deterministic model of packet delay. In *SIGCOMM*. 283–294.

LAI, K. AND BAKER, M. 2001. Nettimer: A tool for measuring bottleneck link bandwidth. In *Proceedings of the USENIX Symposium on Internet Technologies and Systems*.

LILLIEFORS, H. W. 1969. On the kolmogorov-smirnov test for the exponential distribution with mean unknown. *J. Amer. Stat. Assoc. 64*, 325 (Mar.), 387–389.

LUCANTONI, D., NEUTS, M., AND REIBMAN, A. 1994. Methods for performance evaluation of vbr video traffic models. *IEEE/ACM Trans. Networking 2*, 176–180.

MAGLARIS, B., ANASTASSIOU, D., P. SEN, G. K., AND ROBBINS, J. D. 1988. Performance models of statistical multiplexing in packet video communications. *IEEE Trans. Comm. 36*, 7, 834–844.

MAH, B. 1997. An empirical model of HTTP network traffic. In *Proceedings of the IEEE Infocom*. IEEE, Kobe, Japan, 592–600.

MAH, B., SHOLANDER, P., MARTINEZ, L., AND TOLENDINO, L. 1998. Ipb; an internet protocol benchmark using simulated traffic. In *Proceedings of MASCOTS '98*. IEEE, Montreal, Canada.

MAH, B. A. 1999. Pchar: Child of pathchar, presented at the DOE NGI testbed workshop, Berkeley, CA, 21 july 1999.

MASSEY AND JR., F. J. 1951. The Kolmogorov-Smirnov test of goodness of fit. *J. Amer. Stat. Assoc. 46*, 253 (Mar.), 68–78.

MATHIS, M. AND MAHDAVI, J. 1996. Diagnosing Internet congestion with a transport layer performance tool. In *Proceedings of INET '96*. Montreal.

MCCREARY, S. AND CLAFFY, K. 2000. Trends in wide area ip traffic patterns: A view from Ames Internet exchange. *13th ITC Specialist Seminar*, 1–11.

MELAMED, B. AND SENGUPTA, B. 1992. Tes modeling of video traffic.

MINSHALL, G. 1997. Tcpdpriv, http://ita.ee.lbl.gov/html/contrib/tcpdpriv.html.

NIKOLAIDIS, I. AND AKYILDIZ, I. 1992. Source characterization and statistical multiplexing in atm networks. *Tech. Rep. GIT-CC 92-24, Georgia Tech.*

NLANR. 2001. Pma long traces archive, http://pma.nlanr.net/traces/long/.

PAXSON, V. 1994. Empirically derived analytic models of wide-area TCP connections. *IEEE/ACM Trans. Netw. 2*, 4, 316–336.

PAXSON, V. 1997. Measurements and analysis of end-to-end Internet dynamics. In *Ph.D. thesis, University of California, Berkeley*.

PAXSON, V. 1999. End-to-end Internet packet dynamics. *IEEE/ACM Trans. on Netw. 7*, 3, 277–292.

PAXSON, V. 2000. Internet Traffic Archive, http://www.acm.org/sigcomm/ita/.

POSTEL, J. AND REYNOLDS, J. 1985. Rfc959.txt, file transfer protocol (FTP).

SEN, P., MAGLARIS, B., RIKLI, N.-E., AND ANASTASSIOU, D. 1989. Models for packet switching of variable-bit-rate video sources. *IEEE J. Selected Areas Comm. 7*, 5, 865–869.

SMIRNOV, N. 1948. Table for estimating the goodness of fit of empirical distributions. *Annals Math. Stat. 19*, 2 (June), 279–281.

SMITH, F. D., CAMPOS, F. H., JEFFAY, K., AND OTT, D. 2001. What TCP/IP protocol headers can tell us about the web. In *SIGMETRICS/Performance*. Cambridge, MA, 245–256.

VARDI, Y. 1996. Network tomography: Estimating source-destination traffic intensities from link data. *J. Amer. Stat. Assoc. 91*, 433 (Mar.), 365–377.

WILLINGER, W., PAXSON, V., AND TAQQU, M. 1998. Self-similarity and heavy-tails: Structural modeling of network traffic. In *Self-Similarity and Heavy-Tails: Structural Modeling of Network Traffic, in A Practical Guide To Heavy Tails: Statistical Techniques and Applications*, R. J. Adler, R. E. Feldman, and M. S. Taqqu, Eds. ISBN 0-8176-3951-9. Birkh auser, Boston, 1998.

YUKSEL, M., SIKDAR, B., VASTOLA, K. S., AND SZYMANSKI, B. 2000. Workload generation for ns simulations of wide area net works and the Internet. In *Proceedings of Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS) part of Western Multi-Conference (WMC)*. San Diego, CA, 93–98.

ZHANG, Y., DUFFIELD, N., PAXSON, V., AND SHENKER, S. 2001. On the constancy of Internet path properties. In *Proceedings of ACM SIGCOMM Internet Measurement Workshop 2001*. San Francisco Bay Area.

Received October 2002; accepted December 2002